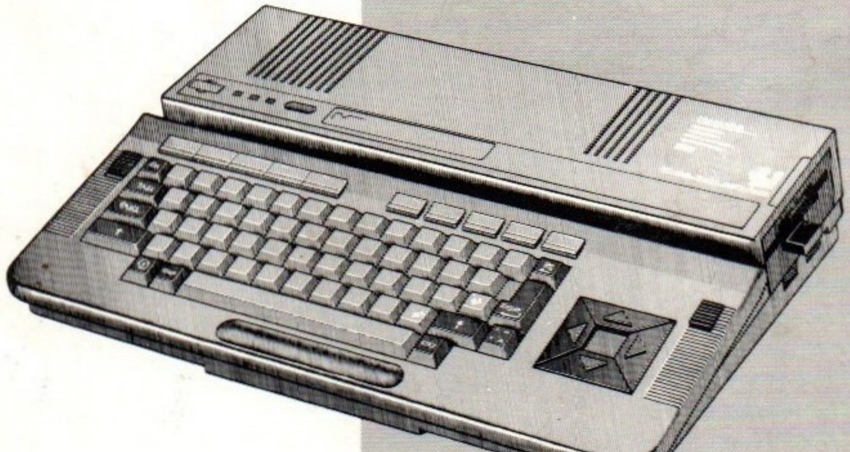
PHILIPS

REFERENCE MANUAL



Dit boek bevat een compleet overzicht van alle commando's voor uw MSX2 computer en moet gebruikt worden in combinatie met uw MSX BASIC handboek

MSX 2

PHILIPS

MSX2 UITBREIDING OP MSX-BASIC

HANDLEIDING

Deze handleiding is een uitgave van Philips Export B.V. te Eindhoven, Nederland.

© Philips Export B.V.

Alle rechten voorbehouden. Niets uit deze handleiding mag worden vermenigvuldigd, in welke vorm dan ook, zonder uitdrukkelijke, schriftelijke toestemming van de uitgever.

MSX en MSX2 zijn handelsmerken van Microsoft Corporation

1. KENMERKEN VAN MSX2-BASIC

MSX2-BASIC is een aanvulling op MSX-BASIC en is compatibel met MSX-BASIC. Dat betekent dat alle instructies van MSX-BASIC ook gebruikt kunnen worden met MSX2-BASIC. De instructies van MSX-BASIC worden besproken en uitgelegd in de handleiding die u bij uw computer hebt gekregen.

Een aantal instructies van MSX-BASIC zijn voor MSX2-BASIC uitgebreid, namelijk de volgende: BASE, BLOAD, BSAVE, CIRCLE, COLOR, COPY, DRAW, LINE, LOCATE, PAD, PAINT, POINT, POS, PRESET, PSET, PUT SPRITE, SCREEN, VDP, VPEEK, VPOKE en WIDTH.

Naast deze uitgebreide instructies bevat MSX2-BASIC ook veel nieuwe instructies, namelijk: CALL MEMINI, CALL MFILES, CALL MKILL, CALL MNAME, COLOR=, COLOR=NEW, COLOR=RESTORE, COLOR SPRITE, COLOR SPRITE$, GET DATE, GET TIME, SET ADJUST, SET BEEP, SET DATE, SET PAGE, SET PASSWORD, SET PROMPT, SET SCREEN en SET TIME.

1. TEKST Met MSX2-BASIC passen maximaal 80 tekens op één regel van het scherm.
Zie de instructies LOCATE, POS, SCREEN en WIDTH.

2. KLEUREN Met MSX2-BASIC kan het palet van elk van de 16 kleuren worden gedefinieerd. Dit betekent dat u 512 verschillende combinaties kunt definiëren. Afhankelijk van de schermmodus kunt u op één scherm 2, 4, 16 of 256 verschillende kleuren gebruiken.
Zie de instructies COLOR, COLOR=, COLOR=NEW en COLOR=RESTORE.

3. GRAPHICS Het oplossend vermogen van het scherm is in MSX2-BASIC

maximaal 512 x 212 pixels, afhankelijk van de schermmodus. In het videogeheugen kunnen verscheidene grafische schermen worden opgeslagen. Eén gedeelte van het scherm kan gekopieerd worden naar een ander gedeelte van het scherm, naar het geheugen of naar de diskette, en andersom natuurlijk.
Zie de instructies CIRCLE, COPY, DRAW, LINE, PAINT, POINT, PRESET, PSET, SCREEN en SET PAGE.

4. VIDEO-GEHEUGEN MSX2-BASIC heeft een groter videogeheugen. Dit heeft geleid tot een verandering van enkele instructies die betrekking hebben op directe toegang tot het videogeheugen.
Zie de instructies BASE, BLOAD, BSAVE, VDP, VPOKE en VPEEK.

5. RAM-DISK Een gedeelte van het geheugen kan als een diskette worden gebruikt. Dat betekent dat niet de diskette maar dit gedeelte van het geheugen wordt gebruikt om gegevens sequentieel weg te schrijven of in te lezen. Voordat u toegang hebt tot de RAM-DISK moet u de instructie CALL MEMINI gebruiken.
Zie de instructies CALL MEMINI, CALL MFILES, CALL MKILL en CALL MNAME.

6. KLOK-CHIP De MSX2-computer bevat een klok-chip die de datum en tijd bijhoudt nadat die eenmaal met SET DATE en SET TIME zijn ingesteld. Het is ook mogelijk om bepaalde startwaarden, bijvoorbeeld een wachtwoord, vast te leggen in het geheugen van de klok-chip.
Zie de instructies GET DATE, GET TIME, SET ADJUST, SET BEEP, SET DATE, SET PASSWORD, SET PROMPT, SET SCREEN, SET TIME en SET TITLE.

7. SPRITES Met MSX2-BASIC is het mogelijk om maximaal 8 sprites op één regel van het scherm te hebben, afhankelijk van de schermmodus. U kunt de kleur van iedere lijn van de sprite afzonderlijk definiëren.
Zie de instructies COLOR SPRITE, COLOR SPRITE$, PUT SPRITE en SCREEN.

8. MUIS, TRACKBALL, LICHTPEN MSX2-BASIC kan de status en de positie van een muis, trackball of lichtpen inlezen.
Zie de instructie PAD.

2. EXTRA INSTRUCTIES VAN MSX2-BASIC

BASE(<X>)

Werkt op dezelfde manier als bij MSX-BASIC, maar <X> kan tussen 0 en 44 liggen. INT(<X>/5) stelt de schermmodus vast. <X> MODE 5 stelt het type tabel vast.

Wanneer een basisadres van een tabel wordt opgevraagd, dan is <X> geldig als de waarde tussen 0 en 44 ligt. Wanneer een basisadres wordt gewijzigd, dan is <X> geldig als de waarde tussen 0 en 19 ligt. Dit betekent dat alleen in tekstmodus 1 en 2 en in grafiekmodus 1 en 2 de basisadressen van de tabellen gewijzigd kunnen worden. Wanneer een basisadres in grafiekmodus 1 wordt gewijzigd, dan worden de basisadressen van de tabellen van grafiekmodus 4, 5, 6, 7 en 8 ook gewijzigd.

BLOAD "<dev>:<bestandsnaam>",S

Werkt op dezelfde manier als bij MSX-BASIC, maar wanneer de schermmodus de grafiekmodus 4, 5, 6 of 7 is, dan wordt de inhoud van de actieve pagina geladen (zie de instructie SET PAGE).

BSAVE "<dev>:<bestandsnaam>",<startadres>,<eindadres>,S

Werkt op dezelfde manier als bij MSX-BASIC, maar wanneer de schermmodus de grafiekmodus 4, 5, 6 of 7 is, dan wordt de inhoud van de actieve pagina opgeslagen (zie de instructie SET PAGE).

CALL MEMINI[(<X>)]

Reserveert een gedeelte van het geheugen als een RAM-DISK. <X> geeft de grootte van de RAM-DISK aan en moet ofwel 0 zijn (dat wil zeggen dat er geen RAM-DISK wordt gebruikt), ofwel een geheel getal zijn boven de 1023. De werkelijke omvang van de RAM-DISK wordt als volgt berekend: (INT((<X>-1023)/256))+1)x256. De standaardwaarde van <X> is 32767. CALL MEMINI(0) heft een bestaande RAM-DISK op. Een CALL MEMINI-instructie wist de bestaande inhoud van de RAM-DISK.

Zonder CALL MEMINI-instructie kan de RAM-DISK niet worden gebruikt. Nadat CALL MEMINI is uitgevoerd, kunnen de instructies CALL MFILES, CALL MKILL en CALL MNAME worden gebruikt. De volgende instructies kunnen dan worden gebruikt om bestanden in te lezen van of weg te schrijven naar de RAM-DISK: SAVE, LOAD, RUN, MERGE, OPEN, CLOSE, PRINT#, PRINT# USING, INPUT#, LINE INPUT#, INPUT$#, EOF, LOC en LOF. Ze worden op dezelfde manier gebruikt als bij MSX-Disk-BASIC, alleen wordt nu in plaats van de naam van de diskette-eenheid, MEM gebruikt. Als er nog geen RAM-DISK is ingesteld met CALL MEMINI of als een RAM-DISK is opgeheven met CALL MEMINI(0), dan zal het gebruik van deze instructies de fout "Disk offline" opleveren.
Voorbeeld :

```
10 REM
20 REM xxxxxxxxxx
30 REM
_MEMINI
SAVE "MEM:PROG.BAS"
```

CALL MFILES
Geeft de namen van de bestanden op de RAM-DISK en laat zien hoeveel ruimte nog beschikbaar is. Zie de instructie CALL MEMINI.
Voorbeeld :

```
10 REM
20 REM xxxxxxxxxx
30 REM
_MEMINI
SAVE "MEM:PROG.BAS"
_MFILES
```

CALL KILL("<bestandsnaam>")
Verwijdert een bestand dat is opgeslagen op de RAM-DISK. Zie de instructie CALL MEMINI.
Voorbeeld :

```
10 REM
20 REM xxxxxxxxxx
30 REM
_MEMINI
SAVE "MEM:PROG.BAS"
_MKILL("PROG.BAS")
```

CALL MNAME("<bestandsnaam-1>" AS "<bestandsnaam-2>")

Wijzigt de naam van een bestand dat is opgeslagen op de RAM-DISK. De oorspronkelijke bestandsnaam is <bestandsnaam-1> en de nieuwe naam voor dat bestand is <bestandsnaam-2>. Zie de instructie CALL MEMINI.

Voorbeeld :

```
10 REM
20 REM xxxxxxxxxx
30 REM
_MEMINI
SAVE "MEM:PROG.BAS"
_MNAME("PROG.BAS" AS "MYFILE.BAS")
```

CIRCLE [STEP](<x,y>),<r>[,<kleur>][,<begin hoek>][,<einde hoek>][,<verhouding>]

Werkt op dezelfde manier als bij MSX-BASIC, maar de schermcoördinaten van het centrum van de cirkel kunnen worden veranderd. In de grafiekmodus 1, 2, 3, 4 en 7 moet de X-coördinaat een geheel getal zijn tussen 0 en 255; in de grafiekmodus 5 en 6 moet de X-coördinaat een geheel getal zijn tussen 0 en 511. In de grafiekmodus 1, 2 en 3 moet de Y-coördinaat een geheel getal zijn tussen 0 en 191; in de grafiekmodus 4, 5, 6 en 7 moet de Y-coördinaat een geheel getal zijn tussen 0 en 211.

In de grafiekmodus 1, 2, 3, 4 en 6 moet het kleurnummer een geheel getal zijn tussen 0 en 15; in de grafiekmodus 5 moet het kleurnummer een geheel getal zijn tussen 0 en 3 en in de grafiekmodus 7 moet het kleurnummer een geheel getal zijn tussen 0 en 255 (zie de instructie COLOR).

COLOR [<voorgrondkleur>][,<achtergrondkleur>][,<randkleur>]

Definieert de kleur van het scherm (zie de instructie CIRCLE voor het gebruik van de kleurnummers).
Welke kleuren nu eigenlijk bij welke nummers van het kleurenpalet horen (behalve in grafiekmodus 7) ziet u in de volgende kleurenpalet-tabel:

Kleuren-palet nummer	*Kleurnaam*	*Intensiteit*		
		rood	*groen*	*blauw*
0	transparant	0	0	0
1	zwart	0	0	0
2	medium-groen	1	6	1
3	lichtgroen	3	7	3
4	donkerblauw	1	1	7
5	lichtblauw	2	3	7
6	donkerrood	5	1	1
7	cyaan	2	6	7
8	medium-rood	7	1	1
9	lichtrood	7	3	3
10	donkergeel	6	6	1
11	lichtgeel	6	6	4
12	donkergroen	1	4	1
13	magenta	6	2	5
14	grijs	5	5	5
15	wit	7	7	7

De rode, groene en blauwe intensiteiten kunnen worden gewijzigd met de instructie COLOR=.

Grafiekmodus 5

In deze modus kunnen verschillend gekleurde stroken als randkleur worden gebruikt. Als het nummer van de randkleur tussen 0 en 3 ligt, dan heeft de hele rand de kleur zoals u die vindt in de kleurenpalet-tabel.
Als het nummer van de randkleur tussen 16 en 31 ligt, dan bestaat de rand uit kleurenstroken die als volgt gecombineerd zijn:

Randkleur nummer	*Resulteert in een combinatie van palet-kleur*
16	0 en 0
17	0 en 1
18	0 en 2
19	0 en 3
20	1 en 0
21	1 en 1
22	1 en 2
23	1 en 3
24	2 en 0
25	2 en 1
26	2 en 2
27	2 en 3
28	3 en 0
29	3 en 1
30	3 en 2
31	3 en 3

Grafiekmodus 7

In deze modus kan het nummer van de voorgrondkleur, de achtergrondkleur en de randkleur tussen 0 en 255 liggen. De feitelijke kleur wordt als volgt vastgesteld: kleurnummer = 4 x R + 32 x G + B. R, G en B zijn de intensiteiten van rood, groen en blauw. Rood en groen moeten waarden hebben tussen 0 en 7. Blauw moet een waarde hebben tussen 0 en 3. Een kleur met rode, groene en blauwe intensiteiten van respectievelijk 7, 5 en 2 zullen kleurnummer 190 hebben (= 4 x 7 + 32 x 5 + 2).

```
10 SCREEN 8
20 OPEN "GRP:" FOR OUTPUT AS #1
30 PRESET (0,0)
40 FOR I=0 TO 255:COLOR I:PRINT
#1,CHR$(219);:NEXT
50 FOR I=0 TO 2000:NEXT
60 CLOSE
70 END
RUN
```

COLOR=(<X>),<XX>,<YY>,<ZZ>)

Wijzigt de rode, groene en blauwe intensiteiten van de kleurenpalet-tabel. **<X>** is het kleurenpalet-nummer en moet een geheel getal zijn tussen 0 en 15. **<XX>** is de rode intensiteit en moet een geheel getal zijn tussen 0 en 7. **<YY>** is de groene intensiteit en moet een geheel getal zijn tussen 0 en 7. <ZZ> is de blauwe intensiteit en moet ook een geheel getal zijn tussen

0 en 7. Deze instructie kan worden gebruikt in elke schermmodus, behalve in grafiekmodus 7.
Het kleurenpalet-nummer 0 staat gewoonlijk voor transparant. Dit betekent dat deze kleur niet zichtbaar is, ongeacht de intensiteiten van rood, groen en blauw. Het nummer 0 kan als ieder ander nummer worden gebruikt door middel van de volgende instructie: VDP(9)=VDP(9)OR&H20. Om nummer 0 terug te zetten op transparant, gebruikt u de volgende instructie: VDP(9)=VDP(9)AND&HDF.
Voorbeeld :

```
10 SCREEN 4
20 LINE (10,10)-(100,100),2,BF
30 FOR I=0 TO 2000:NEXT
40 COLOR=(2,6,3,6)
50 FOR I=0 TO 2000:NEXT
60 END
RUN
```

COLOR[=NEW]
Zet de rode, groene en blauwe intensiteiten van de kleurenpalet-tabel terug op hun standaardwaarde. Deze standaardwaarden zijn beschreven bij de instructie COLOR.
Voorbeeld :

```
10 SCREEN 0:WIDTH 40:COLOR 15
20 COLOR=(15,1,1,2)
30 PRINT "XXX"
40 FOR I=0 TO 2000:NEXT
50 COLOR=NEW
60 PRINT "YYY"
70 END
RUN
```

COLOR=RESTORE
Stelt de rode, groene en blauwe intensiteiten van de kleurenpalet-tabel in op de waarden die in het videogeheugen zijn opgeslagen. Deze instructie is handig wanneer de gegevens voor de kleurenpalet-tabel zijn geladen via de instructie BLOAD....S en de intensiteiten van rood, groen en blauw in de kleurenpalet-tabel moeten worden gewijzigd door de zojuist geladen gegevens. De kleurenpalet-tabel is in het videogeheugen als volgt opgeslagen:

Schermmodus	*Locatie in videogeheugen*
tekstmodus 1 (40 tekens)	&H0400 - &H0420
tekstmodus 1 (80 tekens)	&H0F00 - &H0F20
tekstmodus 2	&H2020 - &H2040
grafiekmodus 1	&H2020 - &H2040
grafiekmodus 2	&H2020 - &H2040
grafiekmodus 3	&H2020 - &H2040
grafiekmodus 4	&H7680 - &H76A0
grafiekmodus 5	&H7680 - &H76A0
grafiekmodus 6	&HFA80 - &HFAA0
grafiekmodus 7	&HFA80 - &HFAA0

In de grafiekmodus 4, 5, 6 en 7 kan meer dan één pagina worden gebruikt (zie de instructie SET PAGE). De volgende berekening geeft het videogeheugenadres van de kleurenpalet-tabel voor de verschillende pagina's:

Schermmodus	*Berekening*
grafiekmodus 4	paginanummer x &H08000 + &H7680
grafiekmodus 5	paginanummer x &H08000 + &H7680
grafiekmodus 6	paginanummer x &H10000 + &HFA80
grafiekmodus 7	paginanummer x &H10000 + &HFA80

Voorbeeld :

```
10 SCREEN 0:WIDTH 40:COLOR 15
20 COLOR=(15,0,7,0)
30 PRINT "XYZ"
40 BSAVE "COL1.PIC",&H400,&H420,S
50 COLOR=NEW
60 FOR I=0 TO 2000:NEXT
70 BLOAD "COL1.PIC",S
80 COLOR=RESTORE
90 END
RUN
```

COLOR SPRITE(<X>)=<Y>
Stelt de kleur in van een opgegeven sprite. Deze instructie kan alleen worden gebruikt in grafiekmodus 3, 4, 5, 6 en 7. <**X**> is het nummer van de sprite zoals dat geregistreerd is in de variabele **SPRITE$(X)**. <**Y**> moet een geheel getal zijn tussen 0 en 111. Wanneer <**Y**> een waarde tussen 0 en 15 is, verwijst hij naar het nummer op het kleurenpalet.
Als u 32 optelt bij het kleurenpalet-nummer van <**Y**>, krijgt het kleurenpalet-nummer de volgende functie: "negeer botsingen van sprite".
Wanneer in dit geval de positie van de sprite samenvalt (botst)

met de positie van een andere sprite, dan springt het programma niet naar de subroutine die opgegeven is in de instructie ON SPRITE GOSUB.
Als u 64 optelt bij het kleurenpalet-nummer van <Y>, krijgt het kleurenpalet-nummer de volgende functie: "negeer prioriteit en botsingen van sprite en gebruik een logische OR-bewerking voor sprite-kleuren".
Wanneer in dit geval de positie van de sprite samenvalt (botst) met de positie van een andere sprite, dan springt het programma niet naar de subroutine die opgegeven is in de instructie ON SPRITE GOSUB. De kleur van deze sprite zal dan het resultaat zijn van een logische OR-bewerking van de kleurennummers van beide sprites.
Kleuren van sprites kunnen worden ingesteld met de instructies COLOR SPRITE, COLOR SPRITE$ of PUT SPRITE. De kleurenpalet-nummers van de laatst uitgevoerde COLOR SPRITE, COLOR SPRITE$ of PUT SPRITE zijn geldig. Wanneer u de kleur van de sprite instelt met de instructies COLOR SPRITE of COLOR SPRITE$, dan moet het kleurenpalet-nummer van de PUT SPRITE-instructie worden weggelaten.
Voorbeeld :

```
10 SCREEN 5,0
20 B$=""
30 FOR I=1 TO 8:READ A:B$=B$+CHR$(A):
NEXT
40 SPRITE$(0)=B$
50 COLOR SPRITE(0)=12
60 FOR I=0 TO 212:PUT SPRITE 0,(I,I),
,0:NEXT
70 DATA 24,60,126,255,36,36,66,129
80 END
RUN
```

COLOR SPRITE$(<X>)=<X$>
Wijzigt de kleur van een lijn van de opgegeven sprite. Deze instructie kan alleen worden gebruikt in grafiekmodus 3, 4, 5, 6 en 7. <X> is het nummer van de sprite zoals dat geregistreerd is in de variabele SPRITE$(X). <X$> is een stringexpressie van maximaal 16 tekens. Ieder teken correspondeert met een lijn van de sprite. Wanneer de tekencode van <X$> tussen 0 en 15 ligt, dan heeft X$ betrekking op een kleurenpalet-nummer.
Als u 32 optelt bij het kleurenpalet-nummer van <X$>, krijgt het kleurenpalet-nummer de volgende functie: "negeer botsingen van sprite".

Wanneer in dit geval de positie van de sprite samenvalt (botst) met de positie van een andere sprite, dan springt het programma niet naar de subroutine die opgegeven is in de instructie ON SPRITE GOSUB.
Als u 64 optelt bij het kleurenpalet-nummer van <**X$**>, krijgt het kleurenpalet-nummer de volgende functie: "negeer prioriteit en botsingen van sprite en gebruik een logische OR-bewerking voor sprite-kleuren".
Wanneer in dit geval de positie van de sprite samenvalt (botst) met de positie van een andere sprite, dan springt het programma niet naar de subroutine die opgegeven is in de instructie ON SPRITE GOSUB. De kleur van deze sprite zal dan het resultaat zijn van een logische OR-bewerking van de kleurennummers van beide sprites.
Als u 128 optelt bij het kleurenpalet-nummer van <**X$**>, dan wordt de lijn van de sprite 32 punten naar links geschoven.
Zie de instructie COLOR SPRITE.
Voorbeeld :

```
10 SCREEN 5,0
20 B$=""
30 FOR I=1 TO 8:READ A:B$=B$+CHR$(A):
NEXT
40 SPRITE$(0)=B$
50 PUT SPRITE 0,(100,100),,0
60 FOR I=0 TO 2000:NEXT
70 COLOR SPRITE$(0)=CHR$(12)+CHR$(1)+
CHR$(130)
80 FOR I=0 TO 2000:NEXT
90 DATA 24,60,126,255,36,36,66,129
100 END
RUN
```

COPY (<X1>,<Y1>)-(<X2,Y2>)[,<XX>] TO (<X3>,<Y3>)[,<YY>[,<bewerking>]]
COPY (<X1>,<Y1>)-(<X2,Y2>)[,<XX>] TO <array>
COPY (<X1>,<Y1>)-(<X2,Y2>)[,<XX>] TO "[<dev>:]<bestandsnaam>"
COPY <array>[,<richting>] TO (<X3>,<Y3>)[,<YY>[,<bewerking>]]
COPY "[<dev>:]<bestandsnaam>"[,<richting>] TO (<X3>,<Y3>)[,<YY>[,<bewerking>]]
COPY <array> TO "[<randapparaat>:]<bestandsnaam>"
COPY "[<dev>:]<bestandsnaam>" TO <array>
Kopieert een gedeelte van een grafiekscherm naar een ander gedeelte van het scherm, naar een array of naar een diskettebestand. Deze instructie kan alleen worden gebruikt in grafiekmodus 4, 5, 6 en 7.

<**X1**> staat voor de **X**-coördinaat van het beginpunt van het brongebied op het scherm. <Y1> staat voor de Y-coördinaat van het beginpunt van het brongebied op het scherm.
<**X2**> staat voor de **X**-coördinaat van het eindpunt van het brongebied op het scherm. <Y2> staat voor de Y-coördinaat van het eindpunt van het brongebied op het scherm.
<**X3**> staat voor de **X**-coördinaat van het beginpunt van het doelgebied op het scherm. <Y3> staat voor de Y-coördinaat van het beginpunt van het doelgebied op het scherm.
Zie de instructie CIRCLE voor informatie over schermcoördinaten.
In grafiekmodus 4, 5, 6 en 7 kan meer dan één pagina worden ingesteld (zie de instructie SET PAGE). <**XX**> is het bronpaginanummer en <YY> is het doelpaginanummer. Als <**XX**> en/of <YY> niet zijn opgegeven, dan gaat de computer uit van de actieve pagina.
<richting> geeft aan in welke richting het kopiëren (de overdracht) zal plaatsvinden. Hiervoor moet een waarde tussen 0 en 3 worden ingevuld, die de volgende betekenis heeft.

Nummer van richting	*Overdracht*
0	van linksboven naar rechtsonder
1	van rechtsboven naar linksonder
2	van linksonder naar rechtsboven
3	van rechtsonder naar linksboven

<array> is een variabele naam. Hij staat voor een reeks van numerieke variabelen en moet voldoende ruimte hebben om de gegevens uit het schermgedeelte te kunnen bevatten. De grootte van de array kan als volgt worden berekend:
INT((<pixel>*(ABS(X2-X1)+1)*(ABS(Y2-Y1)+1)+7)/8)+4
<pixel> is 4 in grafiekmodus 4, 6 en 7; <pixel> is 2 in grafiekmodus 5.
<bewerking> betekent een logische bewerking tussen het kleurnummer van het brongedeelte van het scherm (BK) en het kleurnummer van het doelgedeelte van het scherm (DK). De uitkomst van deze logische bewerking is het kleurenpaletnummer (K) van het doelgedeelte na de uitvoering van deze COPY-instructie. <bewerking> kan een van de volgende woorden zijn:

Woord	*Logische bewerking*
XOR	K = NOT(BK) x DK + BK x NOT(DK)
OR	K = BK + DK
AND	K = BK X DK
PSET	K = BK
PRESET	K = NOT(BK)
TXOR	Hetzelfde als XOR maar de kleur transparant heeft geen effect.
TOR	Hetzelfde als OR maar de kleur transparant heeft geen effect.
TAND	Hetzelfde als AND maar de kleur transparant heeft geen effect.
TPSET	Hetzelfde als PSET maar de kleur transparant heeft geen effect.
TPRESET	Hetzelfde als PRESET maar de kleur transparant heeft geen effect.

Als er geen bewerking is opgegeven, dan gaat de computer uit van de bewerking PSET.
<dev> kan alleen maar een diskette-eenheid zijn. Als er geen <dev> is opgegeven, dan gaat de computer uit van de standaarddiskdrive (meestal drive A).
Voorbeeld :

```
10 SCREEN 6
20 X1=50:Y1=50:X2=100:Y2=0
30 CIRCLE (X1,Y1),40,3
40 PAINT (X1,Y1),3
50 AA=INT((4*(ABS(X2-X1)+1)*
(ABS(Y2-Y1)+1)+7)/8)+4
60 DIM A(AA)
70 COPY (X1,Y1)-(X2,Y2) TO A
80 FOR I=1 TO 2000:NEXT
90 COPY A,2 TO (150,150)
100 FOR I=1 TO 2000:NEXT
110 END
RUN
```

DRAW <string>
Werkt op dezelfde manier als bij MSX-BASIC, maar de kleurmogelijkheden zijn groter geworden (zie de instructie CIRCLE).

GET DATE <X$>[,A]
Haalt de datum op uit de klok-chip. De notatie van de datum in de <**X$**> string kan een van de volgende zijn: MM/DD/JJ of

DD/MM/JJ of JJ/MM/DD, afhankelijk van de taalversie van MSX2. DD is de datum, MM is de maand en JJ is het jaar. (Van het jaar worden alleen de tientallen gegeven, dus bijvoorbeeld 86 en niet 1986.) Als u ook de optionele parameter A opgeeft, wordt de datum beschouwd als alarmdatum. In dat geval zet het systeem alleen de datum (DD) in de <**X$**> string.
Voorbeeld :

```
GET DATE PR$:PRINT PR$
```

GET TIME <X$>[,A]
Haalt de tijd op uit de klok-chip. De notatie van de tijd in de <**X$**> string is de volgende: UU:MM:SS. UU is het uur, MM zijn de minuten en SS zijn de seconden. Als u ook de optionele parameter A opgeeft, wordt de tijd beschouwd als alarmtijd. In dat geval zet het systeem alleen het uur (UU) en de minuten (MM) in de <**X$**> string.
Voorbeeld :

```
GET TIME PR$:PRINT PR$
```

LINE [[STEP](<x1,y1>)]-[STEP](<x2,y2>)[,<kleur>][,B[F]] [,<bewerking>]
Werkt op dezelfde manier als bij MSX-BASIC, maar de schermcoördinaten of de kleur kunnen anders zijn (zie de instructie CIRCLE). Er mag ook een logische bewerking worden uitgevoerd (zie de instructie COPY).

LOCATE [<X>][,<Y>][,<cursor aan/uit>]
Werkt op dezelfde manier als bij MSX-BASIC, maar de X-coördinaat kan tussen 0 en 80 liggen, afhankelijk van het WIDTH-statement.

PAD(<X>)
Geeft de status van een tekenbord, lichtpen, muis of trackball.
Wanneer <**X**> een waarde heeft tussen 0 en 3, dan wordt aangenomen dat een tekenbord is aangesloten op joystick-aansluiting 1.
Wanneer <**X**> een waarde heeft tussen 4 en 7, dan wordt aangenomen dat een tekenbord is aangesloten op joystick-aansluiting 2.
Wanneer <**X**> een waarde heeft tussen 8 en 11, dan wordt aangenomen dat er een lichtpen is aangesloten. Voordat u gegevens van de lichtpen inleest, moet u eerst controleren of PAD(8) als resultaat -1 geeft.
Wanneer <**X**> een waarde heeft tussen 12 en 15, dan wordt aangenomen dat een muis of trackball is aangesloten op joy-

stick-aansluiting 1. Zorg ervoor dat u eerst **PAD(12)** uitvoert, voordat u gegevens inleest van de muis of trackball die aangesloten is op joystick-aansluiting 1.

Wanneer <**X**> een waarde heeft tussen 16 en 19, dan wordt aangenomen dat een muis of trackball is aangesloten op joystick-aansluiting 2. Zorg ervoor dat u eerst **PAD(16)** uitvoert, voordat u gegevens inleest van de muis of trackball die aangesloten is op joystick-aansluiting 2.

Als <**X**> 0 of 4 is, dan wordt de status van het tekenbord gegeven. Wanneer het tekenbord is aangeraakt, dan is het resultaat van deze functie -1; wanneer het tekenbord niet is aangeraakt, dan is het resultaat 0.

Wanneer <**X**> 1 of 5 is, dan wordt de **X**-coördinaat van het tekenbord gegeven.

Wanneer <**X**> 2 of 6 is, dan wordt de **Y**-coördinaat van het tekenbord gegeven.

Wanneer <**X**> 3 of 7 is, dan wordt de status van de schakelaar op het tekenbord gegeven. Het resultaat hiervan is -1 wanneer de schakelaar is ingedrukt; wanneer de schakelaar niet is ingedrukt, dan is het resultaat 0.

Wanneer <**X**> 8 is, dan wordt de status van de lichtpen gegeven. Het resultaat hiervan is -1 wanneer de **X**- en **Y**-coördinaten van de lichtpen beschikbaar zijn; wanneer deze coördinaten niet beschikbaar zijn, dan is het resultaat 0.

Wanneer <**X**> 9 is, dan wordt de **X**-coördinaat van de lichtpen gegeven.

Wanneer <**X**> 10 is, dan wordt de **Y**-coördinaat van de lichtpen gegeven.

Wanneer <**X**> 11 is, dan wordt de status van de schakelaar op de lichtpen gegeven. Het resultaat hiervan is -1 wanneer de schakelaar is ingedrukt; wanneer de schakelaar niet is ingedrukt, dan is het resultaat 0.

Wanneer <**X**> 12 of 16 is, dan wordt de status van de muis of de trackball gegeven. Het resultaat van deze functie is altijd -1.

Wanneer <**X**> 13 of 17 is, dan wordt de **X**-coördinaat van de muis of trackball gegeven.

Wanneer <**X**> 14 of 18 is, dan wordt de **Y**-coördinaat van de muis of trackball gegeven.

Wanneer <**X**> 15 of 19 is, dan zal het systeem altijd 0 als resultaat geven.

Voorbeeld :

```
10 SCREEN 2
20 AA=0
30 IF PAD(0)=0 THEN 20
40 X=PAD(1):Y=PAD(2)
50 IF AA=0 THEN PSET (X,Y) ELSE LINE
-(X,Y)
60 AA=1
70 GOTO 30
RUN
```

PAINT [STEP](<x,y>)[,<tekenkleur>][,<randkleur>]
Werkt op dezelfde manier als bij MSX-BASIC, maar de schermcoördinaten en de kleuren kunnen anders zijn, en de kleuren kunnen worden veranderd (zie de instructie CIRCLE).

POINT(<x,y>)
Werkt op dezelfde manier als bij MSX-BASIC, maar de schermcoördinaten kunnen anders zijn (zie de instructie CIRCLE).

POS(0)
Werkt op dezelfde manier als bij MSX-BASIC, maar het resultaat kan tussen 0 en 79 liggen.

PRESET [STEP](<x,y>)[,<kleur>][,<bewerking>]
Werkt op dezelfde manier als bij MSX-BASIC, maar de schermcoördinaten en de kleur kunnen anders zijn (zie de instructie CIRCLE). Er mag ook een logische bewerking worden uitgevoerd op de kleuren (zie de instructie COPY).

PSET [STEP](<x,y>)[,<kleur>][,<bewerking>]
Werkt op dezelfde manier als bij MSX-BASIC, maar de schermcoördinaten en de kleur kunnen anders zijn (zie de instructie CIRCLE). Er mag ook een logische bewerking worden uitgevoerd op de kleuren (zie de instructie COPY).

PUT SPRITE <prioriteitsnummer>[,[STEP](<x,y>)][,<kleur>]
Werkt op dezelfde manier als bij MSX-BASIC, maar de kleur van de sprite kan anders zijn (zie de instructie COLOR SPRITE).
In grafiekmodus 2 en 3 kunnen slechts vier sprites op één regel van het scherm worden gezet. In grafiekmodus 4, 5, 6 en 7 kunnen acht sprites op één regel van het scherm worden gezet.

SCREEN [<X>[,<Y>[,<Z>[<XX>[,<YY>[,<ZZ>]]]]]

Stelt de schermmodus in, de afmeting van de sprite, de toetsklik, de baudrate voor de datarecorder, het type printer dat wordt gebruikt en de weergavemodus. De waarde van SCREEN moet een geheel getal zijn tussen 0 en 8, en heeft de volgende betekenis:

0 = tekstmodus 1 (40 x 24 tekens of 80 x 24 tekens, afhankelijk van de WIDTH-instructie)

1 = tekstmodus 2 (32 x 24 tekens)
2 = grafiekmodus 1 (256 x 192 pixels)
3 = grafiekmodus 2 (64 x 48 pixelblokken)
4 = grafiekmodus 3 (256 x 192 pixels)
5 = grafiekmodus 4 (256 x 212 pixels)
6 = grafiekmodus 5 (512 x 212 pixels)
7 = grafiekmodus 6 (512 x 212 pixels)
8 = grafiekmodus 7 (256 x 212 pixels)

Wanneer <X> niet wordt opgegeven, dan wordt de laatst ingestelde schermmodus gebruikt.
De standaardwaarde van <X> is 0.
De volgende kleurencombinaties zijn op één scherm beschikbaar:

schermmodus	0: 2 van 512 kleuren
schermmodus	1: 2 van 512 kleuren
schermmodus	2: 16 van 512 kleuren
schermmodus	3: 16 van 512 kleuren
schermmodus	4: 16 van 512 kleuren
schermmodus	5: 16 van 512 kleuren
schermmodus	6: 4 van 512 kleuren
schermmodus	7: 16 van 512 kleuren
schermmodus	8: 256 kleuren

Zie de instructie COLOR.
De grafische instructies CIRCLE, COLOR SPRITE, COLOR SPRITE$, COPY, DRAW, LINE, PAINT, PSET, PRESET, ON SPRITE GOSUB, SPRITE/ON/OFF/STOP, POINT en PUT SPRITE kunnen alleen worden gebruikt wanneer een grafiekmodus is geactiveerd.
<Y> bepaalt de grootte van de sprites die gebruikt moeten worden. De waarde van <Y> moet een geheel getal zijn tussen 0 en 3, en heeft de volgende betekenis:

0 = kleine sprites (8 x 8 pixels)
1 = kleine sprites, vergroot tot 16 x 16 pixels
2 = grote sprites (16 x 16 pixels)
3 = grote sprites, vergroot tot 32 x 32 pixels

Wanneer <Y> niet wordt opgegeven, dan wordt de laatst ingestelde sprite-grootte gebruikt.
De standaardwaarde van <Y> is 0.
In grafiekmodus 2 en 3 kunnen slechts vier sprites op één regel van het scherm worden gezet. In grafiekmodus 4, 5, 6 en 7 kunnen acht sprites op één regel van het scherm worden gezet.
<Z> geeft aan of de toetsen een klik geven wanneer ze worden ingedrukt. De waarde van <Z> is ofwel 0 ofwel 1, en heeft de volgende betekenis:

0 = geen toetsklik
1 = toetsklik

Wanneer <Z> niet wordt opgegeven, dan wordt laatste instelling gebruikt.
De standaardwaarde van <Z> is 1.
<XX> geeft de baudrate voor de datarecorder. De waarde van <XX> is ofwel 1 ofwel 2, en heeft de volgende betekenis:

1 = 1200 baud
2 = 2400 baud

Wanneer <XX> niet wordt opgegeven dan wordt de laatst ingestelde baudrate gebruikt.
De standaardwaarde van <XX> is 1.
<YY> vertelt de computer of er al of niet een specifieke MSX-printer wordt gebruikt. De waarde van <YY> is ofwel 0 ofwel 1, en heeft de volgende betekenis:

0 = MSX-printer
1 = geen MSX-printer

Wanneer <YY> niet wordt opgegeven dan wordt de laatst ingestelde printerkeuze gebruikt.
De standaardwaarde van <YY> is 0.
<ZZ> bepaalt de weergavemodus. De waarde van <ZZ> moet een geheel getal zijn tussen 0 en 3, en heeft de volgende betekenis:

0 = normaal
1 = interlaced
2 = normale even/oneven alternerende weergave
3 = interlaced even/oneven alternerende weergave

In weergavemodus 2 en 3 moet het schermpaginanummer oneven zijn (zie de instructie SET PAGE). In beide gevallen wordt de schermpagina (oneven nummer) en pagina (oneven nummer min 1) alternerend weergegeven.
Wanneer <ZZ> niet wordt opgegeven, dan wordt de laatst ingestelde weergavemodus gebruikt.
De standaardwaarde van <ZZ> is 0.
Na de SCREEN-instructie worden alle sprites van het scherm gehaald.

Voorbeeld :

```
10 COLOR 15,4,4:FOR I=2 TO 8
20 SCREEN I
30 LINE (32,32)-(200,132),3
40 FOR K=0 TO 3000:NEXT
50 NEXT I
60 END
RUN
```

SET ADJUST (<X>,<Y>)

Wijzigt de oorspronkelijke plaats van de positie links boven op het scherm (0,0). <**X**> moet een geheel getal zijn tussen -7 en 8. Als <**X**> een positief getal is, dan wordt de linker bovenhoek van het scherm naar rechts verplaatst. Als <**X**> een negatief getal is, dan wordt de linker bovenhoek van het scherm naar links verplaatst. <**Y**> moet een geheel getal zijn tussen -7 en 8. Als <**Y**> een positief getal is, dan wordt de linker bovenhoek van het scherm omhoog verplaatst. Als <**Y**> een negatief getal is, dan wordt de linker bovenhoek van het scherm omlaag verplaatst.

De waarden van <**X**> en <**Y**> worden opgeslagen in het klokgeheugen. Dat betekent dat deze instructie automatisch wordt uitgevoerd als u de computer opnieuw aanzet.

Voorbeeld :

```
SET ADJUST (8,-7)
```

SET BEEP <X>,<Y>

Wijzigt de toonhoogte en het volume van de pieptoon. <**X**> staat voor de toonhoogte, en moet een geheel getal zijn tussen 1 en 4. <**Y**> staat voor het volume, en moet ook een geheel getal zijn tussen 1 en 4; 4 geeft het maximumvolume. De waarden van <**X**> en <**Y**> worden opgeslagen in het klokgeheugen. Dat betekent dat deze instructie automatisch wordt uitgevoerd als u de computer opnieuw aanzet.

Voorbeeld :

```
SET BEEP 3,2
```

SET DATE <X$>[,A]

Stelt in de klok-chip de datum in. De notatie van de datum in de <**X$**> string kan een van de volgende zijn: MM/DD/JJ of DD/MM/JJ of JJ/MM/DD, afhankelijk van de taalversie van MSX2. DD is de datum, MM is de maand en JJ is het jaar. (Van het jaar worden alleen de tientallen gegeven, dus bijvoorbeeld 86 en niet 1986.)

Als u ook de optionele parameter A opgeeft, wordt de datum

beschouwd als alarmdatum. In dat geval wordt in de klokchip alleen de datum (DD) ingesteld.
De <**X$**> string wordt opgeslagen in het klokgeheugen. Dat betekent dat het systeem de datum automatisch bijhoudt.
Voorbeeld :

```
SET DATE "01/05/86"
```

SET PAGE <X>,<Y>
Kiest een schermpagina als de actieve pagina, en kiest een schermpagina als de weergavepagina. Deze instructie kan alleen worden gebruikt in grafiekmodus 4, 5, 6 en 7. <**X**> is het nummer van de weergavepagina en moet een geheel getal tussen 0 en 3 zijn, afhankelijk van de gekozen grafiekmodus. <**Y**> is het nummer van de actieve pagina en moet ook een geheel getal tussen 0 en 3 zijn, afhankelijk van de gekozen grafiekmodus.
Een actieve pagina is de pagina waarop invoer/uitvoerinstructies van kracht zijn. Een weergavepagina is de pagina die op het scherm staat. Het totale aantal pagina's varieert al naar gelang de gekozen grafiekmodus (met een videogeheugen van 128K):

Grafiekmodus	*Totale aantal pagina's*
4	0 tot 3
5	0 tot 3
6	0 tot 1
7	0 tot 1

De standaardwaarde van <**X**> en <**Y**> is 0.

Voorbeeld :

```
10 COLOR 15,4,4
20 OPEN "GRP:" FOR OUTPUT AS #1
30 SCREEN 5
40 SET PAGE 0,0:COLOR 15
50 PRESET (32,32)
60 PRINT #1,"000"
70 SET PAGE 0,1:COLOR 1
80 PRESET (32,16)
90 PRINT #1,"111"
100 FOR I=1 TO 5
110 SET PAGE 1,1
120 FOR J=1 TO 1000:NEXT J
130 SET PAGE 0,0
```

```
140 FOR J=1 TO 1000:NEXT J
150 NEXT I
160 CLOSE
170 END
RUN
```

SET PASSWORD <X$>
Stelt een wachtwoord in. <**X$**> is de wachtwoord-string en mag maximaal 255 tekens lang zijn. De <**X$**> string wordt opgeslagen in het klokgeheugen. Dit betekent dat het wachtwoord wordt bewaard: wanneer de computer opnieuw wordt aangezet, verschijnt het woord "Password:" ("Wachtwoord:") op het scherm. De computer wacht nu tot de gebruiker het juiste wachtwoord heeft ingetypt. Als het ingevoerde wachtwoord niet hetzelfde is als het wachtwoord dat in het klokgeheugen is opgeslagen, vraagt de computer gewoon opnieuw om het wachtwoord.
U kunt echter maar één van de instructies SET PASSWORD, SET PROMPT en SET TITLE gebruiken. Wanneer twee of meer van deze instructies worden uitgevoerd, dan is alleen de laatste geldig. Deze drie instructies gebruiken namelijk hetzelfde gedeelte van het klokgeheugen.
Voorbeeld :

```
SET PASSWORD "ZZZ"
```

SET PROMPT <X$>
Verandert de aanwijzing van MSX2-BASIC. <**X$**> is de nieuwe aanwijzing; ze mag maximaal zes tekens lang zijn. De nieuwe aanwijzing komt in de plaats van de aanwijzing "Ok", die de standaardaanwijzing van MSX2-BASIC is. De <**X$**> string wordt opgeslagen in het klokgeheugen. Dat betekent dat deze instructie automatisch wordt uitgevoerd als u de computer opnieuw aanzet.
U kunt echter maar één van de instructies SET PASSWORD, SET PROMPT en SET TITLE gebruiken. Wanneer twee of meer van deze instructies worden uitgevoerd, dan is alleen de laatste geldig. Deze drie instructies gebruiken namelijk hetzelfde gedeelte van het klokgeheugen.
Voorbeeld :

```
SET PROMPT "YYY"
```

SET SCREEN
Slaat verschillende parameters die betrekking hebben op het scherm, op in het klokgeheugen. De volgende parameters kunnen met deze instructie in het klokgeheugen worden opgeslagen:

schermmodus	0 of 1
breedte	1 tot 80
paletkleur van voorgrond	0 tot 15
paletkleur van achtergrond	0 tot 15
paletkleur van rand	0 tot 15
afbeelden van functietoetsen	ON of OFF
toetsklik	ON of OFF
printermodus	MSX of niet-MSX
baudrate datarecorder	1200 of 2400
weergavemodus	0 tot 4

Doordat deze parameters in het klokgeheugen worden opgeslagen, zijn ze onmiddellijk beschikbaar wanneer u de computer opnieuw aanzet.
Voorbeeld:

```
10 SCREEN 0:WIDTH 80:COLOR 1,15:KEY OFF
20 SET SCREEN
30 END
RUN
```

SET TIME <X$>[,A]
Stelt in de klok-chip de tijd in. De notatie van de tijd in de <**X$**> string is de volgende: UU:MM:SS. UU is het uur, MM zijn de minuten en SS zijn de seconden. Als u ook de optionele parameter A opgeeft, wordt de tijd beschouwd als alarmtijd. In dat geval zet het systeem alleen het uur (UU) en de minuten (MM) in het klokgeheugen.
De <**X$**> string wordt opgeslagen in het klokgeheugen. Dat betekent dat het systeem de tijd automatisch bijhoudt.
Voorbeeld:

```
SET TIME "10:53:06"
```

SET TITLE <X$>[,<Y>]
Bepaalt een titel die op het scherm verschijnt wanneer u het systeem aanzet. <**X$**> is de titel-string; ze mag maximaal zes tekens lang zijn. Wanneer de titel-string inderdaad zes tekens lang is, zal het systeem - na het afbeelden van het titelscherm -wachten tot er een toets wordt ingedrukt. <**Y**> is het kleurenpalet-nummer van het titelscherm en moet een geheel getal zijn tussen 1 en 4. De <**X$**> string en de waarde van <**Y**> worden opgeslagen in het klokgeheugen. Dat betekent dat de titel automatisch op het scherm verschijnt als u de computer opnieuw aanzet. U kunt echter maar één van de instructies SET PASSWORD, SET PROMPT en SET TITLE gebruiken. Wanneer twee of meer van deze instructies worden uitgevoerd, dan is alleen de laatste geldig. Deze drie instructies

gebruiken namelijk hetzelfde gedeelte van het klokgeheugen.
Voorbeeld :

```
SET TITLE "XXX",2
```

VDP<X>
Bevat de inhoud van de VDP-registers. **<X>** is het registernummer en moet een geheel getal zijn tussen -9 en -1 of tussen 0 en 8 of tussen 9 en 24 of tussen 33 en 47.
De inhoud van register 8 en de statusregisters -9 tot -1 kan niet worden gewijzigd. Ze kan alleen worden opgeroepen.
Voorbeeld :

```
10 FOR I=0 TO 8
20 A=VDP(I)
30 B$="00000000"+BIN$(A)
40 PRINT RIGHT$(B$,8)
50 NEXT
60 END
RUN
```

VPEEK(<X>)
Werkt op dezelfde manier als bij MSX-BASIC, maar de waarde van **<X>** moet hier tussen 0 en 65535 liggen. Bij een grafiekmodus tussen 4 en 7 wordt het absolute adres berekend als **<X>** + startadres van de actieve pagina. Door middel van de volgende berekening kunt u het startadres van de verschillende pagina's vinden:

Schermmodus	*Berekening*
grafiekmodus 4	paginanummer x &H08000
grafiekmodus 5	paginanummer x &H08000
grafiekmodus 6	paginanummer x &H10000
grafiekmodus 7	paginanummer x &H10000

VPOKE <X>,<Y>
Werkt op dezelfde manier als bij MSX-BASIC, maar de waarde van **<X>** moet hier tussen 0 en 65535 liggen. Bij een grafiekmodus tussen 4 en 7 wordt het absolute adres berekend als **<X>** + startadres van de actieve pagina. Door middel van de volgende berekening kunt het startadres van de verschillende pagina's vinden:

Schermmodus	*Berekening*
grafiekmodus 4	paginanummer x &H08000
grafiekmodus 5	paginanummer x &H08000
grafiekmodus 6	paginanummer x &H10000
grafiekmodus 7	paginanummer x &H10000

WIDTH <X>

Werkt op dezelfde manier als bij MSX-BASIC, maar de waarde van <X> moet hier tussen 1 en 80 liggen. Wanneer de schermmodus 0 is, dan kiest de instructie WIDTH automatisch de modus voor tekenweergave. Als <X> kleiner dan of gelijk is aan 40, dan krijgt u grote tekens op het scherm. Als <X> groter is dan 40, dan krijgt u kleine tekens op het scherm.